International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Sentiment Analysis using a Multinomial LR-LSTM Model

Seema Rani[1], Jai Bhagwan[2]*, Sanjeev Kumar [3], Yogesh Chaba[4], Sunila Godara[5], Sumit Sindhu[6]

Submitted: 07/02/2024 **Revised**: 15/03/2024 **Accepted**: 21/03/2024

Abstract: Sentiment analysis (SA) refers to a technique utilized to ascertain the emotional state conveyed in information or text. It involves categorizing the text into three classes: positive, negative, or neutral. For instance, when someone says "the aqi of the city is good," they are expressing a positive opinion about the aqi of a specific place, while the statement "the aqi is bad" reflects the opposite. The introduction of social media increased the amount of content on the internet of sentiment data. Users on various social media platforms have been able to offer their opinions on various products, services, etc. These opinions are often expressed on social media in the form of movie reviews, product reviews, user comments, posts, etc. In light of this context, one of the captivating research areas in Natural Language Processing (NLP) is Twitter sentiment analysis. The paper proposes a stacked Multinomial-LR-LSTM model for the classification of tweets into three classes. Tweets are re-annotated using Text Blob. Twitter Sentiment dataset was used for experiments with accuracy of 97%.

Keywords: *Deep Learning, Sentiment Analysis, Machine Learning, LSTM, Classification*

1. Introduction

Social media posts encompass a wide array of content, including personal perspectives, thoughts on various subjects, current news, and internet-related matters. Given the abundance of opinions, sentiment analysis becomes a valuable tool. Organizations can leverage the data available on social platforms to enhance the effectiveness of their products and services. This eliminates the need for conventional methods like surveys or opinion polls to gather user feedback. Instead, businesses can swiftly and effortlessly inquire about their customers' thoughts on social media, and subsequently analyse the responses to identify their preferences, dislikes, and areas for improvement [1]. In contemporary politics, political parties are utilizing sentiment analysis to enhance their attractiveness. An illustrative instance of sentiment analysis in action is evident in the prime ministerial election. By examining tweets related to the election, it becomes feasible to gauge the number of voters who hold positive, negative, or neutral sentiments towards him. As a result, sentiment analysis proves to be a valuable tool for analysts. Typically, this analysis is accomplished through a blend of NLP and Machine Learning (ML) techniques [2]. With the emergence of deep language models, more complicated domains of data, such as news articles, where authors express their thoughts and feelings with a reduced amount of explicitness, may be analysed. This study uses the Twitter Sentiment dataset that contains 162980 tweets, 44.3% are positive, 21.7% negative and 33.8% neutral. In this study, Text Blob is used to reassess the tweets annotations. Many studies typically focus on classifying tweets into positive and negative labels for sentiment analysis, disregarding neutral label. They argue that neutral texts are less likely to provide insights into sentiment polarity compared to the clear-cut positive or negative sentiment classes. However, it is crucial to acknowledge the neutral label since some tweets may not be able to convey positive or negative sentiment [3].

Furthermore, relying solely on positive and negative classifications will not accurately classify neutral tweets. For instance, consider "This TV Show is Unpredictable." Without context, it is impossible to determine whether the show is good or bad, indicating the absence of any emotion. By limiting classification to two classes, future predictions will be inaccurate for such texts. Hence, the neutral class, representing a lack of emotion, should be treated as a distinct and separate category rather than

[1]*Assistant Professor, Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology, Hisar, Haryana, India. 125001.*
Email: seemarajput1028@gmail.com, Orchid ID: 0000-0001-7741-6715
[2]*Assistant Professor, Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology, Hisar, Haryana, India. 125001.*
**Corresponding Author's Email: jaihisar007@gmail.com, Orchid ID: 0000-0002-8708-4029*
[3]*Professor, Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology, Hisar, Haryana, India. 125001.*
sanjukhambra@yahoo.co.in
[4]*Professor, Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology, Hisar, Haryana, India. 125001.*
yogeshchaba@yahoo.com
[5]*Professor, Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology, Hisar, Haryana, India. 125001.*
sunilagodara@gmail.com
[6]*Reserach Scholar, Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology, Hisar, Haryana, India. 125001.*

being perceived as a state between positive and negative sentiments. The paper proposes a stacked Multinomial-LR-LSTM model for the classification of tweets into three classes. Multinomial-LR will be used as it allows for more than two variables for output. LSTM is very good for remembering large dependencies when using a large dataset. Therefore, two LSTM models are combined to improve accuracy.

2. Related Work

A SA model was presented by Lal Khan et al. [4] for the classification of English and Roman-Urdu literature. They came up with a system that integrated two methods of deep learning. Long Short-Term Memory (LSTM) was employed to preserve long-term dependencies and Convolutional Neural Network (CNN) was used for feature extraction. Following that, machine learning classifiers were fed the LSTM output. The approach suggested by Gen Li et al. [5] is aimed at sentiment extraction from Chinese text. By using a hybrid approach, they were able to anticipate sentiment patterns by capturing fundamental emotional feelings. According to experimental results, their model performed better than other models that were already in use and showed better generalization abilities. A classification model for Roman-Urdu and English text was presented by Lal Khan et al. [4]. Two deep learning techniques were merged into a framework that they created. They employed Long Short-Term Memory (LSTM) to preserve long-term dependencies and Convolutional Neural Network (CNN) for feature extraction. Machine learning classifiers were subsequently fed the LSTM output. A model on sentiment extraction from Chinese text was presented by Gen Li et al. [5]. The researchers employed a hybrid technique to predict sentiment trends by attempting to capture fundamental emotional feelings. The model they developed performed better than other models that were already in use, according to experimental results. Meylan Wongkar et al. [8] suggested a Naive Bayes (NB) algorithm-based SA model. Their model tried to categorize different emotional states. When they examined the effectiveness of NB, SVM, and K-Nearest Neighbors (KNN), NB had the best accuracy (75%), followed by KNN (73%), and SVM (64%). In order to identify sentiments, U. Sehar et al. [9] provided a thorough framework that combined text, audio, and graphic replies. They found valuable trends by utilizing a special dataset that had 1372 phrases. They increased the polarity detecting capability to 95% using their method.
A combined approach that blended CNN and Bidirectional-LSTM with word embedding was presented by S. Tam et al. [10]. Their model performed better than others, with an accuracy of 91.13%. A model for Bengali natural language processing (NLP) that was created by Al Amin et al. [11] and modified from the VADER model allowed for the recognition of Bengali sentiment polarities. To achieve better results, the writers also used Bengali boosting words and stemming. The main goal, according to Davcheva E et al. [12], was to analyze shifts in mental states and get a better understanding of diverse circumstances. The study's conclusions showed that, depending on the circumstance, sentiment score tended to be either positive or negative. Three layered LSTM models make up the stacked ensemble model that Gaye, B. et al. [13] suggested. This layered approach's output was input into a classifier for logistic regression (LR). Using Text Blob, the writers reassessed the default attitudes. Their algorithm outperformed the default feelings model by a substantial margin, with an astounding 99% accuracy rate in tweet analysis.

3. Materials and Methods

3.1 Dataset Description

The models undergo training and evaluation using the Twitter Sentiment dataset [14], which comprises 162,980 tweets. This dataset is composed of three distinct classes: positive, negative, and neutral sentiments. It consists of two columns, namely "text" and "label." The distribution of the classes is as follows: 44.3% positive, 21.7% negative, and 33.8% neutral.

To further refine the dataset, the tweets are re-evaluated using Text Blob, resulting in a re-labelling process. The re-annotated tweets are assigned numerical values, where 1 represents positive sentiment, 0 represents negative sentiment, and 2 represents neutral sentiment.

3.2 Data Pre-Processing

The majority of data is gathered from web platforms such as twitter. The data is either unstructured or semi-structured. Pre-processing plays a significant role in eliminating the noise and re-organizing the data. Effective pre-processing techniques can reduce the feature set by 30-50%, retaining only the pertinent features. Large datasets result in longer training times and less accurate predictions due to stop words, punctuations and data not relevant to study. Therefore, pre-processing is necessary to efficiently use computational power, to efficiently train ML models, and produces more accurate predictions.

- Data Cleaning: Punctuation and numbers have no effect on the tweet's sentiment; hence they are not required for sentiment analysis. Similar to that, usernames are likewise unimportant for classifying text's emotion.
- Eliminating stop words: Stop words that are “most common words in the text” contribute

nothing but computer overhead. Stop words are removed in this step.

- Lowercasing text: Tweets are transformed to lower case because case matters and words that seem the same in upper or lowercase will be treated differently by ML algorithms, this reduces the effectiveness of the classifier.
- Stemming and Lemmatization: The process of stemming is “converting a word into a root word or stem”. E.g., Words “eats” “eaten” “eating” are stemmed into “eat”. The Lemmatization approach examines the meaning of the word, whereas the stemming technique merely considers the form of the word.

3.3 TextBlob

TextBlob is a Python library that is designed for conducting various NLP tasks [15]. When utilizing TextBlob, two outputs are obtained: a sentiment score and a subjectivity score. Sentiment score falls within the range of "-1.0" to "+1.0," where "-1.0" indicates negative and "+1.0" signifies positive sentiment [16]. Integrated TextBlob with several ML classifiers and authors of [13] integrated TextBlob with ML as well as DL classifiers but excluded the neutral class. Both studies show that TextBlob boosts the model's performance.

3.4 Term Frequency Inverse Document Frequency (TF-IDF)

TF-IDF [17] uses a vocabulary to get its features. TF-IDF has better performance for the extraction of features as accuracy of TF-IDF increases when the “frequency of occurrence of a word” is increased and its accuracy decreases when the “frequency of occurrence of a word” is decreased. TF-IDF consists of two elements: TF and IDF. TF calculates the “frequency of a word” in a document. It is calculated as the likelihood of discovering a text term within a document. IDF displays a word's frequency or rarity across the corpus. Rare words can be found using IDF. Weight is computed by the formula [26] discussed by equation 1.

$$W_{i,j} = TF_{i,j}\left(\frac{N}{D_{f,t}}\right) \qquad (1)$$

where TFi,j denote the frequency of occurrence of term t in document d, N represents the total number of documents, and Df,t represents the count of documents that contain term t.

3.5 Machine Learning and Deep Learning Models

- K-Nearest Neighbour (KNN): KNN is a ML algorithm based on the idea of feature similarity. KNN is an instance-based learning approach that relies on a distance function to calculate the proximity between two instances [18]. KNN is also known as the lazy learner algorithm.
- Naïve Bayes (NB): NB classifier is based on the probability or assumptions that features are independent of each other. There is mainly three naïve classifier Gaussian Naïve Bayes (GNB) which works for continuous data values, Bernoulli Naïve Bayes (BNB) works for binary, Multinomial Naïve Bayes (MNB) helps in discrete values.
- Logistic Regression (LR): Unlike its name, LR is not a regression method but a classification method [19]. LR predicts whether an event will happen or not, such as “performed the task” or “did not perform the task”. LR provides a probability-based output. The output varies between 0 and 1.
- Decision Tree Classifier (DTC): A DTC is a hierarchical model that uses the branching method for decision. It can predict both continuous and discrete values. Tree learner represents if-then rules, and three basic elements of a decision tree are decision node, branch and leaf nodes.
- AdaBoost (ADB): ADB is an ensemble technique. It employs adaptive boosting for training weak learners [20]. To create a single strong classifier, ADB combines several weak classifiers. Weak learners are trained recursively to duplicate the original data. Weak learners concentrate on problematic outliers.
- Random Forest (RF): RF is a versatile supervised ML algorithm derived from decision trees [21]. It is applicable for both classification and regression tasks. RF consists of multiple decision trees, working together to solve complex problems by averaging the output of these decision trees.
- Convolution Neural Network (CNN): The CNN is one of the finest classifiers with promising results using an array for data storage. In its basic structure, it includes an input, output and multiple hidden layers, including pooling, convolution, and connection layers. CNN is also used for supervised feature extraction.
- Long Short-Term Memory (LSTM) [22] is a variant of RNN with additional capability of conserving relevant information for a long time. LSTM consists of memory units also known as cells and gates. Cells are used to update, preserve and edit the information. Gates are used to decide which data to keep and which

data to discard. LSTM cell equation is given [23][24][25] using equations (2), (3) and (4).

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$f_t = \sigma\left(w_i[h_{t-1}, x_t] + b_f\right) \quad (3)$$

$$O_t = \sigma(w_i[h_{t-1}, x_t] + b_0) \quad (4)$$

Where i_t represents input gate, σ represents sigmoid function, w_i represents weight of gate, h_{t-1} represents output received from previous LSTM gate at t-1 time, x_t represents input at current time, b_i and represents bias.

3.6 Proposed Model

This research proposes a stacked Multinomial-LR-LSTM model for classification of tweets into three classes. In a stacked architecture, multiple deep learning layers are stacked one after another. In the proposed model, 2 LSTM layers are stacked. LSTM requires 3D array input and outputs a 2D array. As 3D input is required, the first LSTM layer returns “output of sequences” rather than a single output to the next LSTM layer. Multiple dense layers are used to improve accuracy. For final classification, Multinomial-LR will be used as three classes are present in the dataset.

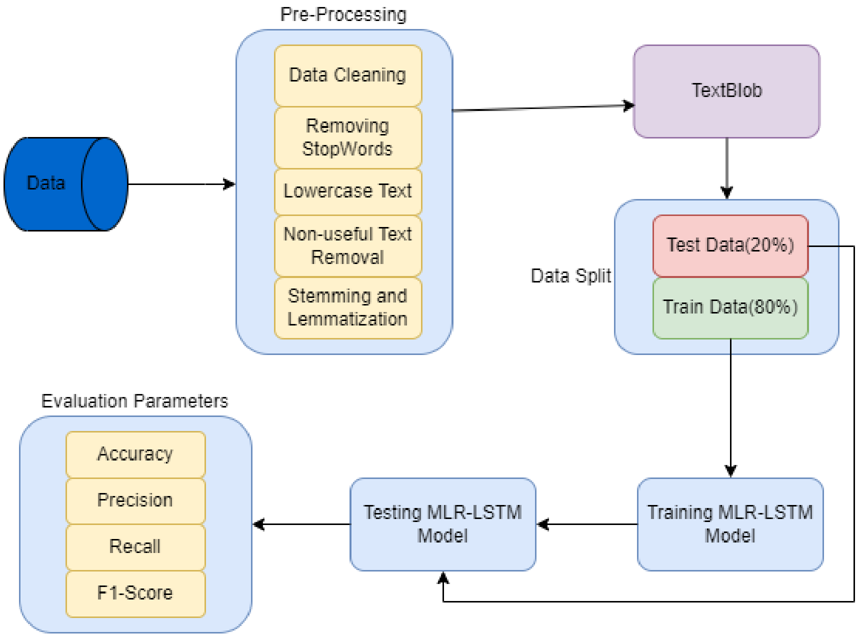

Fig 1. Model Structure of the proposed methodology

Twitter Sentiment dataset is used for experiments. The annotations of tweets in the dataset are reassessed using Text Blob. After re-annotation of tweets, 41.9% of them are positive, 21.4% are negative, and 36.5% are neutral. Then pre-processing is done. Pre-processing steps include data cleaning, removing stop words, lowercasing, non-useful text removal, stemming, and lemmatization. Following data pre-processing, the dataset is divided into training and test sets of 80% and 20% respectively. The model is trained on the training set and subsequently evaluated using various metrics, including accuracy, precision, recall, and F1-score.

4. Results Discussion

This section presents experimental findings and analysis. The ML models use TF-IDF feature extraction for the experiments, which were conducted using both default sentiments as well as sentiments extracted by Text Blob. The DL models use techniques such as CNN, LSTM etc.

4.1 Results of ML Models with Default Sentiment using TF-IDF

The table below presents the outcomes of ML models using default sentiments. The highest accuracy of 88% is obtained by Logistic Regression (LR), which exhibits a precision of 88%, recall of 87%, and an F1-score of 87%. The KNN model achieves the poorest performance with an accuracy of 54%.

Regarding individual class prediction, Random Forest (RF) performs the best for the negative class, achieving a precision of 90%. For positive classes, LR outperforms other models with a precision of 92%. As for the neutral class, LR also demonstrates the best performance, with a precision of 85%.

Table 1. Results of ML models with Default sentiments using TF-IDF

Classifier	Accuracy	Class	Precision	Recall	F1-score
DTC	0.81	Negative	0.72	0.66	0.69

		Positive	0.83	0.82	0.83
		Neutral	0.83	0.88	0.85
		Macro avg.	0.79	0.79	0.79
RF	0.84	Negative	0.90	0.59	0.71
		Positive	0.82	0.90	0.86
		Neutral	0.84	0.92	0.88
		Macro avg.	0.85	0.82	0.82
LR	0.88	Negative	0.87	0.76	0.81
		Positive	0.92	0.89	0.90
		Neutral	0.85	0.96	0.90
		Macro avg.	0.88	0.87	0.87
ADB	0.76	Negative	0.80	0.56	0.66
		Positive	0.89	0.71	0.79
		Neutral	0.66	0.97	0.78
		Macro avg.	0.78	0.74	0.74
KNN	0.54	Negative	0.60	0.31	0.41
		Positive	0.73	0.42	0.54
		Neutral	0.45	0.84	0.58
		Macro avg.	0.59	0.52	0.51
BNB	0.76	Negative	0.72	0.46	0.56
		Positive	0.75	0.86	0.80
		Neutral	0.79	0.83	0.81
		Macro avg.	0.76	0.72	0.72

4.2 Results of DL Models with Original Sentiment

As indicated in the table below, the CNN-LSTM model exhibited superior performance, outperforming all other models with an accuracy of 87%, along with a precision of 90%, recall of 86%, and an F1-score of 88%. In comparison, both CNN and LSTM individually achieved lower results than the combined CNN-LSTM approach.

For individual class prediction, LSTM proved to be the best performer for the negative and neutral classes, achieving a precision of 88% for the negative class and 92% for the neutral class. On the other hand, CNN outperformed the other models for the positive class with a precision of 95%.

Table 2. Results of DL models with Default sentiments

Classifier	Accuracy	Class	Precision	Recall	F1-score
CNN	0.84	Negative	0.85	0.79	0.82
		Positive	0.95	0.81	0.88
		Neutral	0.89	0.93	0.91
		Macro avg.	0.90	0.84	0.87
CNN-LSTM	0.87	Negative	0.87	0.77	0.82
		Positive	0.92	0.89	0.91
		Neutral	0.90	0.91	0.91
		Macro avg.	0.90	0.86	0.88
LSTM	0.86	Negative	0.88	0.80	0.83
		Positive	0.93	0.89	0.91
		Neutral	0.92	0.87	0.89
		Macro avg.	0.91	0.85	0.88

4.3 Results of Proposed Model with Original Sentiments

The accuracy of the sentiment classification model is enhanced by stacking two LSTM layers. LSTM has the ability for automatically extracting features. As evident from the results presented in the table below, the proposed model surpasses the best DL model with an impressive accuracy of 88%, accompanied by a precision of 91%, recall of 87%, and an F1-score of 89%.

Table 3. Experimental results of Proposed DL model with Default sentiments

Classifier	Accuracy	Class	Precision	Recall	F1-score
MLR-LSTM	0.88	Negative	0.89	0.81	0.85
		Positive	0.94	0.90	0.92
		Neutral	0.91	0.92	0.91
		Macro avg.	0.91	0.87	0.89

4.4 Results of ML Models with TextBlob Sentiments

The table below displays the results of ML models with TextBlob sentiments. The highest accuracy of 94% is attained by the Logistic Regression (LR) model, exhibiting a precision of 94%, recall of 93%, and an F1-score of 93%. On the other hand, the KNN achieves the lowest performance with an accuracy of 56%.

Regarding individual class prediction, Random Forest (RF) and LR perform exceptionally well for negative classes, both attaining a precision of 94%. For positive classes, LR outperforms other models with a precision of 95%. The neutral class is best predicted by the Decision Tree Classifier (DTC) with a precision of 96%.

In general, Logistic Regression (LR) demonstrates the most robust performance across all metrics. The findings indicate that combining the TF-IDF feature extraction technique with TextBlob reassessment of tweets yields superior results, and logistic regression outperforms all other models.

Table 4. Results of ML models with TextBlob sentiments using TF-IDF

Classifier	Accuracy	Class	Precision	Recall	F1-score
DTC	0.91	Negative	0.83	0.80	0.82
		Positive	0.90	0.91	0.90
		Neutral	0.96	0.98	0.97
		Macro avg.	0.90	0.90	0.90
RF	0.90	Negative	0.94	0.72	0.82
		Positive	0.89	0.94	0.91
		Neutral	0.90	0.97	0.93
		Macro avg.	0.91	0.87	0.89
LR	0.94	Negative	0.94	0.86	0.90
		Positive	0.95	0.94	0.95
		Neutral	0.92	0.98	0.95
		Macro avg.	0.94	0.93	0.93
ADB	0.82	Negative	0.86	0.64	0.74
		Positive	0.93	0.76	0.84
		Neutral	0.72	0.99	0.83
		Macro avg.	0.84	0.80	0.80
KNN	0.56	Negative	0.66	0.31	0.42
		Positive	0.75	0.42	0.54
		Neutral	0.48	0.87	0.62
		Macro avg.	0.63	0.53	0.52

BNB	0.80	Negative	0.75	0.53	0.52
		Positive	0.77	0.90	0.83
		Neutral	0.86	0.85	0.85
		Macro avg.	0.79	0.76	0.77

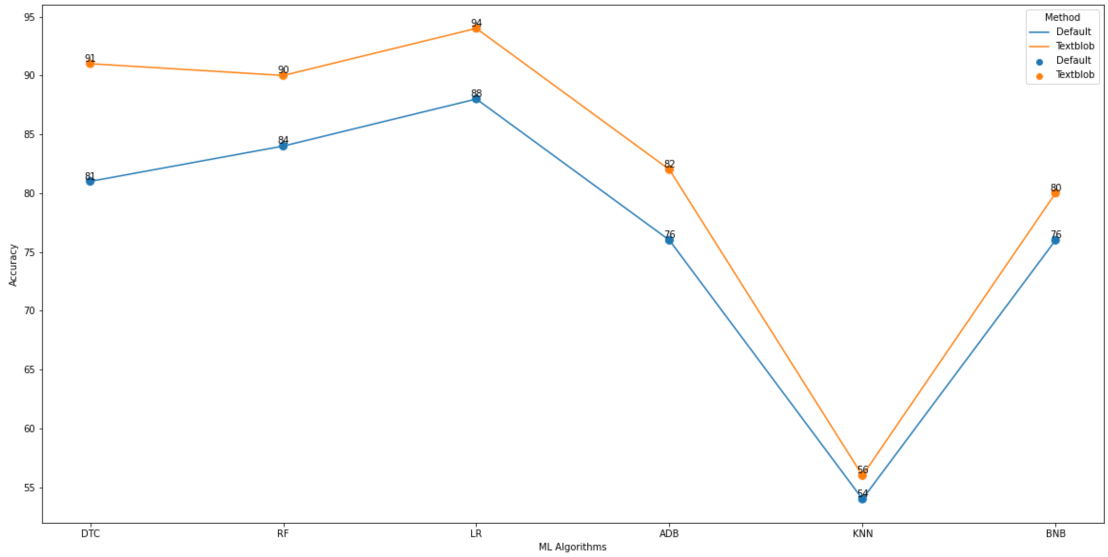

Fig 2. Performance Comparison of ML models with Default and TextBlob sentiment

4.5 Results of DL Models with TextBlob Sentiment

Here, sentiments are reassessed using TextBlob on the Twitter Sentiment dataset. CNN-LSTM and LSTM have a similar accuracy of 96% when annotation of tweets is reassessed using TextBlob. As for individual class prediction, CNN-LSTM was the best of three, with a precision of 97% for the negative class. For the positive class CNN and LSTM achieve precision of 98% and neutral classes, CNN-LSTM and LSTM achieves precision of 99%.

Table 5. Results of DL models with TextBlob sentiments

Classifier	Accuracy	Class	Precision	Recall	F1-score
CNN	0.95	Negative	0.94	0.93	0.94
		Positive	0.98	0.93	0.96
		Neutral	0.97	0.98	0.97
		Macro avg.	0.96	0.95	0.96
CNN-LSTM	0.96	Negative	0.97	0.92	0.94
		Positive	0.97	0.97	0.97
		Neutral	0.98	0.98	0.98
		Macro avg.	0.97	0.96	0.97
LSTM	0.96	Negative	0.96	0.95	0.96
		Positive	0.98	0.97	0.97
		Neutral	0.98	0.98	0.98
		Macro avg.	0.98	0.97	0.97

4.6 Results of Proposed Model with Text Blob Sentiments

Results of the proposed model are shown in the table below. Stacking of learning models is used to improve the accuracy of models. Two LSTM layers are stacked to improve the sentiment classification model's accuracy. Output of stacked LSTM layers is provided to multinomial LR. The model achieves accuracy of 97%.

Table 6. Experimental results of Proposed DL model with Text Blob sentiments

Classifier	Accuracy	Class	Precision	Recall	F1-score
MLR-LSTM	0.97	Negative	0.97	0.95	0.96
		Positive	0.98	0.98	0.98
		Neutral	0.99	0.98	0.98
		Macro avg.	0.98	0.97	0.98

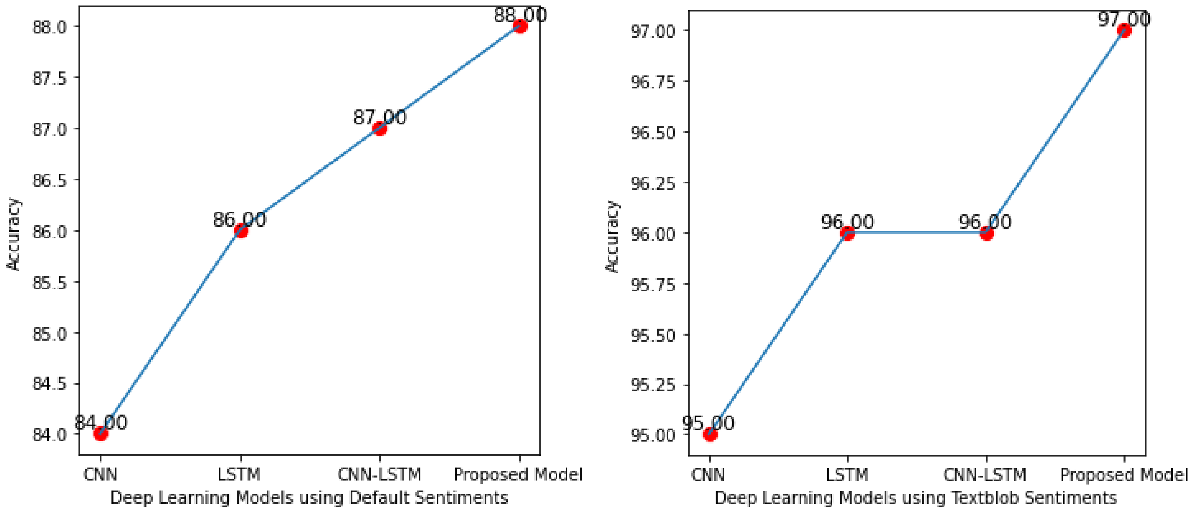

Fig 3. Performance Comparison of Proposed Model with DL Models

4.7 Comparative Analysis of Proposed Model with Other Models

Proposed model outperforms the traditional ML models and DL models in all performance metrics and the proposed model with Text Blob sentiments massively outperforms the model using default sentiments. It is clear that the Text Blob sentiments used in the proposed model significantly exceed the model's use of default sentiments. It is observed that the tweets reassessed by Text Blob have more correlation with textual features of the text than default tweets. Three DL models were used to compare the performance of the proposed model. All three DL models performed somewhat similar to one another, so there was room for improvement. The model is based on LSTM which are networks with input, output and loops in them, LSTM networks are perfect for sequence and pattern learning. There is no fixed number of layers in a neural network, these are decided by looking and trying out what number of layers are perfect for your model and data-set. Multiple dense layers are used to improve accuracy. For final classification, Multinomial-LR will be used as three classes are present in the data-set. Proposed model provides better performance than DL models.

5. Conclusion

In this study, a stacked model of 2-LSTM and Multinomial-LR is proposed. In many studies, the neutral class is often disregarded, but it is essential to incorporate it as some tweets or text may not convey any specific sentiment. Including the neutral class ensures a more accurate classification of text. In this study, the tweets were re-evaluated using the TextBlob library, resulting in three distinct sentiment classes. The reassessed tweets showed a stronger correlation with the tweets.

Multiple ML and DL models were employed to assess the performance of the proposed model. For ML models, the feature extraction technique TF-IDF was used. Both the original sentiment and reassessed sentiments were used in the experiments. The results demonstrate that combining TF-IDF feature extraction with TextBlob reassessment of tweets yields superior outcomes, with logistic regression outperforming all other models. The proposed model exhibited exceptional performance, surpassing all other models, both ML and DL, achieving an impressive accuracy of 97%.

References

[1] Sindhu, S., Kumar, S., & Noliya, A. (2023, March). A Review on Sentiment Analysis using Machine Learning. In 2023 *International Conference on Innovative Data Communication Technologies and Application (ICIDCA)* (pp. 138-142). IEEE.

[2] Khan, M. L., Ittefaq, M., Pantoja, Y. I. M., Raziq, M. M., & Malik, A. (2021). Public engagement model to analyze digital diplomacy on Twitter: A

social media analytics framework. *International Journal of Communication*, 15, 29.

[3] Hamed, A.R., Qiu, R. and Li, D., 2016. The importance of neutral class in sentiment analysis of Arabic tweets. *Int. J. Comput. Sci. Inform. Technol*, *8*, pp.17-31.

[4] Khan, L., Amjad, A., Afaq, K.M. and Chang, H.T., 2022. Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media. *Applied Sciences*, *12*(5), p.2694.

[5] Li, G., Zheng, Q., Zhang, L., Guo, S. and Niu, L., 2020, November. Sentiment infomation based model for Chinese text sentiment analysis. In *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)* (pp. 366-371). IEEE.

[6] Naqvi, U., Majid, A. and Abbas, S.A., 2021. UTSA: Urdu text sentiment analysis using deep learning methods. *IEEE Access*, *9*, pp.114085-114094.

[7] Wang, Y., Huang, G., Li, J., Li, H., Zhou, Y. and Jiang, H., 2021. Refined global word embeddings based on sentiment concept for sentiment analysis. *IEEE Access*, *9*, pp.37075-37085.

[8] Wongkar, M. and Angdresey, A., 2019, October. Sentiment analysis using Naive Bayes Algorithm of the data crawler: Twitter. In *2019 Fourth International Conference on Informatics and Computing (ICIC)* (pp. 1-5). IEEE.

[9] Sehar, U., Kanwal, S., Dashtipur, K., Mir, U., Abbasi, U. and Khan, F., 2021. Urdu Sentiment Analysis via Multimodal Data Mining Based on Deep Learning Algorithms. *IEEE Access*, *9*, pp.153072-153082.

[10] Tam, S., Said, R.B. and Tanriöver, Ö.Ö., 2021. A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification. *IEEE Access*, *9*, pp.41283-41293.

[11] Amin, A., Hossain, I., Akther, A. and Alam, K.M., 2019, February. Bengali vader: A sentiment analysis approach using modified vader. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-6). IEEE.

[12] Davcheva, E., Adam, M. and Benlian, A., 2019. User dynamics in mental health forums–a sentiment analysis perspective.

[13] Gaye, B., Zhang, D. and Wulamu, A., 2021. A Tweet sentiment classification approach using a hybrid stacked ensemble technique. *Information*, *12*(9), p.374.

[14] Twitter Sentiment Dataset: HUSSEIN, SHERIF (2021), "Twitter Sentiments Dataset", Mendeley Data, V1, doi: 10.17632/z9zw7nt5h2.1

[15] Loria, S., 2018. textblob Documentation. *Release 0.15*, *2*(8).

[16] Rupapara, V., Rustam, F., Shahzad, H.F., Mehmood, A., Ashraf, I. and Choi, G.S., 2021. Impact of SMOTE on imbalanced text features for toxic comments classification using RVVC model. *IEEE Access*, *9*, pp.78621-78634.

[17] Yu, B., 2008. An evaluation of text classification methods for literary study. *Literary and Linguistic Computing*, *23*(3), pp.327-343.

[18] Jiang, L., Cai, Z., Wang, D. and Jiang, S., 2007, August. Survey of improving k-nearest-neighbor for classification. In *Fourth international conference on fuzzy systems and knowledge discovery (FSKD 2007)* (Vol. 1, pp. 679-683). IEEE.

[19] Kleinbaum, D.G.; Klein, M.; Pryor, E.R. Logistic Regression: A Self-Learning Text; Springer: New York, NY, USA, 2002.

[20] Zhang, Y., Zhang, H., Cai, J. and Yang, B., 2014, May. A weighted voting classifier based on differential evolution. In *Abstract and Applied Analysis* (Vol. 2014). Hindawi.

[21] Da Silva, N.F., Hruschka, E.R. and Hruschka Jr, E.R., 2014. Tweet sentiment analysis with classifier ensembles. *Decision support systems*, *66*, pp.170-179.

[22] Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, p.132306.

[23] Li, C., Wang, Z., Rao, M., Belkin, D., Song, W., Jiang, H., ... & Xia, Q. (2019). Long short-term memory networks in memristor crossbar arrays. *Nature Machine Intelligence*, 1(1), 49-57.

[24] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv*:1506.00019.

[25] W. Medhat, A. Hassan, and H. Korashy, ''Sentiment analysis algorithms and applications: A survey,'' Ain Shams Eng. J., vol. 5, no. 4, pp. 1093–1113, 2014, doi: 10.1016/j.asej.2014.04.011.

[26] Lubis, A. R., Nasution, M. K., Sitompul, O. S., & Zamzami, E. M. (2021). The effect of the TF-IDF algorithm in times series in forecasting word on social media. *Indones. J. Electr. Eng. Comput. Sci*, 22(2), 976.